

In the Specification

On page 3 after line 6 please insert the following one paragraph.

Figures 40 to 49: Examples of access dependent contents.

On page 4, line 3 please take out the word “www” and put in its stead the words “http://httpd” thereby creating the URL “http://httpd.apache.org”.

At the time of this disclosure, the best mode contemplated by the inventor for carrying out the present invention is an implementation as an enhancement to the publicly-available Apache Web server software (see [\[\[www\]\]http://httpd.apache.org/](http://httpd.apache.org/)). It is expected, however, that once the usefulness of the present invention has become apparent to more people, the present invention will be considered in the design of new system architectures. Because of relative complexities of retrofitting into existing architectures, such as Apache, careful balance has been struck in this disclosure in order to provide a clear and concise yet still full and exact description of the present invention. In order to encourage widespread adaptation, neither a specific programming language nor a specific graphical user interface object library are required.

On page 5, lines 1 and 3 please take out both occurrences of the word “users” and put in their stead “people”.

According to the theory and practice of the present invention, “looks do matter.”matter”. In order to achieve the highest rates of correct decisions by operators in the shortest possible time with least effort, what the operator sees is of critical importance. Beyond a certain point, quantitative differences in visual appearance become qualitative differences. By analogy: If the font size of text is below a certain size, e.g. below four, then no one can read at all; if the font size is above a certain number, e.g. above thirty, then most [\[\[users\]\]people](#) can read even if they didn’t bring their glasses and would need glasses for regular sized text, e.g. for size ten. A quantitative difference in font size effects a difference in whether [\[\[users\]\]people](#) can read at all, i.e. a qualitative difference.

One page 5, line 6 please take out the words “a user” and put in their stead “an operator”.

According to the theory and practice of the present invention, there are practical limits to the amount of abstract reasoning that should be demanded of ~~a user~~an operator of a system. These limits may be different from person to person; the limits may be different for a one-time effort and for repeated performance. According to the theory and practice of the present invention one can achieve much higher rates of correct decisions by operators by relieving them of the need to perform complex symbolic mental operations.

On page 36, line 7 after the word “omnipresence” please insert the punctuation mark comma “,”.

Layout is important in theory and practice of the present invention. Operator comprehension, and consistency should be increased through consistent and clever layout of user interface elements. One potential problem could be circumstances when there are more items to display than would fit into predetermined display regions, e.g. when there are a large number of users who are allowed to read a document. A commonly employed technique is to use scroll bars. The potential for pervasiveness, for omnipresence, is an important part of the appeal of the present invention. Consequently, minimal use of display space is an essential virtue in practice of the present invention. That said, there are at least three problems with commonly known scroll bars: (1) Scroll bars use display space themselves. That may not be a problem for a single document window which covers most of the display device, but it becomes a problem when several small regions each would need scroll bars of their own. When scroll bars use in excess of e.g. fifty percent of available display space, then there is a problem. (2) Scroll bars can cause abrupt reformatting of contents. Once content has increased to the point that it requires scroll bars, the very appearance of scroll bars takes away some of the display region which previously had been available for contents. That can cause reformatting, i.e. contents to be laid out somewhat differently in the remaining display region. (3) Scroll bars have proven not to fit in with the visual appearance that has been achieved with implementations of the present invention. If operators pay less attention to access control settings because individual, auxiliary, user interface elements divert their attention, then those user interface elements should be considered a problem. Scroll bars are considered integral part of most user interface object platforms, and, for reasons which to explore would exceed the scope of this disclosure, there is little variation being practiced by software that employs them.

On page 41, line 17 take out the word “in” and put in its stead the word “is”.

If the present invention enables people to control who can access what document, what will follow is a greater diversity in access control settings than in environments where access control is more difficult to manipulate. One consequence will be that hyperlinks have a much higher probability to lead a user to a document that the user ~~[[in]]~~is not authorized to access. Here are two examples:

On page 42, line 2 take out the word “on” and put in its stead the word “of”.

The present invention suggests a significant improvement in dealing with these situations, by the use of markup which causes conditional omission of elements ~~[[on]]~~of documents, conditional on access control settings for resources, in general resources other than the document itself, resources which are identified by reference, mostly that would be resources which are referenced in the section of the document which is subject to conditional omission.

On page 42, line 21 take out the words “access:if” and put in their stead “adc:onlyIf”.

More formally defined, the present invention suggests a system for omitting elements from documents as a function of user identity and access control settings which comprises a processing mechanism which processes documents before passing them on, a parsing mechanism in the processing mechanism which parses markup language for elements and attributes, and an access control function mechanism which is activated by the parsing mechanism when a defined access control function attribute, e.g. ~~access:if~~adc:onlyIf=“../internal/customerlist.html”, is encountered. The attribute should provide a value which identifies a resource, e.g. ../internal/customerlist.html. The access control function mechanism should make a determination of access privileges which the user has for that resource, and then should by applying its predetermined function to the access privileges make a decision on whether to omit the element, e.g. a , a <p>, or a , with which the attribute is associated, e.g. to omit if the user is not allowed to read document ../internal/customerlist.html. That decision should be implemented by the processing mechanism, coordinately acting together with the other mechanisms.

On page 44, line 3 insert before the word “One” the sentence “E.g. `adc:onlyIfRefs=“true”` (synonymous to `adc:onlyIfAllRefs=“true”`) or `adc:onlyIfAnyRefs=“true”`.”

It might be considered useful to implement support for an access control function attribute that during processing automatically extracts references from the contents of its element. E.g. `adc:onlyIfRefs=“true”` (synonymous to `adc:onlyIfAllRefs=“true”`) or `adc:onlyIfAnyRefs=“true”`. One disadvantage, however, would be higher computational effort at each access. Another disadvantage would be the need to understand potential contents of elements.

On page 44 after line 5 please insert the following two paragraphs.

As an example, Figure 40 shows underlying markup of a document. Figure 41 shows the document in a WYSIWYG editor, for its author. Figures 42 and 43 show the document as it consequentially is rendered differently for different users.

As another example, Figure 44 shows underlying markup of a document that refers to images. Figure 45 shows the document, including images, for its author. Figure 46 shows diverse access control settings of individual images. Figures 47 to 49 show the document as it consequentially is rendered differently for different users.

On page 45 after line 18 please insert the following one paragraph.

Good implementations of the present invention should function well for communication among people whose native languages are different. Many of its features are language neutral, and GUI implementations should lend themselves well for different localizations (internationalization) nevertheless seamlessly accessing the same resources.

On page 48 after line 2 please insert the following five paragraphs, i.e. four references.

References

Programmer’s Guide to the Java 2D API, Java 2 SDK, Standard Edition, 1.2, Sun Microsystems, Inc.

Scalable Vector Graphics (SVG) 1.1 Specification, W3C Candidate Recommendation
Document Object Model (DOM) Level 2 Specification, W3C Working Draft
Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation